

REMARKS/ARGUMENTS

The Office Action mailed March 22, 2006 has been carefully considered.

Reconsideration in view of the following remarks is respectfully requested.

Claims 1 – 34 are pending in the application. Claims 1, 2, 7 – 9, 11, 14 – 17, 24, and 31 – 34 have been amended. Support for the amendments is found in the specification, drawings, and claims as originally filed. Applicants submit, therefore, that the amendments do not add new matter.

Claim Objections

The Examiner has objected to claims 1, 2, 9, 11, 14 – 17, 24, and 31. In response applicants have amended the claims.

The 35 U.S.C. § 112, Second Paragraph Rejection

Claims 7, 8, 15, 16 and 31-34 were rejected under 35 U.S.C. § 112, second paragraph, as allegedly being indefinite for failing to particularly point out and distinctly claim the subject matter applicant regards as the invention.

In response applicants have amended claims 7, 8, 15, 16 and 31-34 to distinctly claim the subject matter.

The First 35 U.S.C. § 103 Rejection

Claims 1-6, 9, 14, 17-25 and 32-34 were rejected under 35 U.S.C. § 103(a) as being allegedly unpatentable over Anandakumar et al.¹, among which claims 1, 9, 14, 20 and 32 are independent claims. This rejection is respectfully traversed.

Applicants submit that claim 1, as amended is not anticipated or rendered obvious by Anandakumar. Amended claim 1 includes the following limitations.

A process for automatically altering the data rate on a logical channel comprising:

- 1) resetting a flawed packet counter for said logical channel in response to changing noise conditions on the logical channel;
- 2) resetting a total packets received counter for said logical channel;
- 3) receiving a packet on said logical channel, and incrementing the total packet received counter;
- 4) processing error detection data in said packet to determine if there is an error in the packet, and, if so, incrementing said flawed packet counter;
- 5) comparing the count in said total packet counter to a number representing the desired number of packets to be received before a determination of packet loss percentage is made;
- 6) if the number of packets received is less than the desired number, returning to step 3;
- 7) if the number of packets received is equal to or exceeds the desired number of packets received, calculating a packet loss percentage by dividing the number of flawed packets by the total number of packets received;
- 8) comparing the packet loss percentage calculated in step 7 to one or more packet loss thresholds;
- 9) determining if a change in data rate throughput is required based upon the comparison(s) made in step 8;
- 10) if a change in data rate throughput is required, generating a signal indicating the need for a change in data rate for said logical channel.

(Amended claim 1) (Emphasis added)

Applicants respectfully submit that Anandakumar does not teach or suggest the limitation of resetting a flawed packet counter for said logical channel in response to changing noise conditions on the logical channel. Anandakumar discloses the following.

“Operations of add diversity block 1561 of FIG. 15 are illustrated in FIG. 16. A software implementation is illustrated. In FIG. 16, operations commence at BEGIN 1601 and proceed to a step 1605 to initialize a vector STATE having vector element values s (source rate) and d (diversity rate). The values s and d are initialized with initial values $s11$ and $d11$ respectively.

Next in FIG. 16, a loop begins at an input step 1611 to input a newly-arriving QoS datum such as a new RTCP report QoS inverse measure such as the packet loss fraction L . If no RTCP packet is present, operations branch to a RETURN 1614. If an RTCP packet is present, then operations proceed to a decision step 1615.

In the decision step 1615, the value L is compared to determine if L exceeds a first threshold, designated Threshold1 or $Th1$, indicating too much packet loss at the destination. If too much packet loss, then operations proceed from step 1615 to a decision step 1617 to determine whether L also exceeds an even higher level A . If L is less than or equal to A , operations go next to a moderate update step 1621. If L exceeds A , operations go to an aggressive update step 1623.

In step 1621, a vector NEWSTATE is moderately updated in the manner shown in FIG. 1 and intended to improve QoS and likely reduce value L expected subsequently when the destination reports back. NEWSTATE is an intermediate state value holding vector, used intermediately in controlling the values called STATE. NEWSTATE is suitably updated according to any software method selected by the skilled worker, such as by looking up in a table, or executing a CASE statement, or implementing a software state machine, or otherwise. If the source rate can be decreased no further, and the diversity can be increased no further, then step 1621 simply makes NEWSTATE the same as the current state. After step 1621 operations go to step 1651.

In step 1623, vector NEWSTATE is aggressively updated (for example to state $(s32, d32)$) in the manner shown in FIG. 23 and intended to improve QoS and likely reduce value L expected subsequently when the destination reports back. Here again, NEWSTATE is suitably updated according to any software method selected by the skilled worker, such as by looking up in a table, or executing a CASE statement, or implementing a software state machine, or otherwise. If the source rate can be decreased no further, and the diversity can be increased no further, then step 1623 simply makes NEWSTATE the same as the current state. After step 1623 operations go to step 1651.

Some embodiments have thresholds B , C , etc., such that NEWSTATE is moderately updated if $(A \geq F > Th1)$, NEWSTATE is aggressively updated if $(B \geq F > A > Th1)$, and NEWSTATE is even more aggressively updated if $(F > B > A > Th1)$.

If in step 1615, L does not exceed Threshold1, operations proceed to a decision step 1627. Step 1627 determines if either the gatekeeper signal GK is on ($GK=1$) OR a buffer full flag is on ($BFR=1$). If NO, then operations go directly to step 1625, but if YES, operations go to an update step 1629.

In step 1629, vector NEWSTATE is updated (for example to state $(s31, d31)$) in the manner

¹ U.S. Patent No: 6,765,904

shown in FIG. 23 and intended to improve QoS and likely reduce value L expected subsequently when the destination reports back. Here again, NEWSTATE is suitably updated according to any software method selected by the skilled worker, such as by looking up in a table, or executing a CASE statement, or implementing a software state machine, or otherwise. After step 1629, operations go to step 1651.

In decision step 1625, the packet loss fraction value L is compared with a second threshold Threshold2, or Th2. If value L is less than Th2, then QoS has improved or is at a high level already. In such case, operations pass to a step 1631 to update NEWSTATE to increase the source rate. Step 1631 inputs a new estimated steady state overall transmission rate S. As in steps 1621 and 1623, NEWSTATE is suitably updated according to any software method selected by the skilled worker, such as by looking up in a table, or executing a CASE statement, or implementing a software state machine, or otherwise. As indicated in FIG. 1 and FIG. 10, the process need not be simply the reverse of the transitions available in step 1621, and so step 1631 is customized for its own updating purposes. Also, if the source rate can be increased no further, and the diversity can be decreased no further, then step 1631 simply makes NEWSTATE the same as the current state. One embodiment performs aggressive overall transmission rate increase if the ratio R of estimated steady state overall transmission rate S to current overall transmission rate exceeds a threshold Th3 (e.g. 3.0), and performs gradual overall transmission rate increase otherwise. For example, such embodiment uses TCP throughput estimate for new estimated steady state overall transmission rate S. The TCP throughput estimate is suitably that given earlier (5D) as $1.22 \cdot \text{packetsize} / (\text{round-trip delay} \cdot \sqrt{\text{average loss measured during lifetime of the connection}})$.

After step 1631 operations go to a step 1651. If in step 1625, value were greater than or equal to second threshold Th2, then operations go to decision step 1635.

In decision step 1635, the packet loss fraction value L is tested to see if it lies in the range from first threshold Th1 to second threshold Th2 inclusive. If yes, then operations go to a step 1641 wherein intermediate NEWSTATE is filled with the values in the vector STATE. Together with a later step 1651, this operation 1641 maintains the current control state. If the decision in step 1635 is NO (out of range), or when step 1641 is completed, then operations pass to step 1651.

In step 1651 the vector STATE is filled with the values of NEWSTATE. Next in an output step 1661, the values of STATE are output as control signals (s_{ij} , d_{ij}) to the encoder 321 of FIG. 3 and the encoder 1551 of FIG. 15. Steps 1651 and 1661 thus implement transitions like 101 and 413 (see FIG. 4) and other transitions discussed herein. From step 1661 operations pass to a decision step 1671 whether to STOP. If not, then operations loop back to step 1611 and continue repeatedly as discussed above. If STOP is yes, then operations go to END 1681. STOP may be responsive to disconnection of communications with the particular destination, or to power off or to other conditions of a chip or system as desired.

(Anandakumar, col. 36, line 4 – col. 37, line 54)

The Examiner has stated that it would have been obvious based on the cited portion of Anandakumar that the packet loss described would include packet loss due to noise. Applicants respectfully submit that regardless of the cause of the packet loss, Anandakumar does not disclose or suggest the limitation of resetting a flawed packet counter for said logical channel in response to changing noise conditions. Anandakumar makes no reference at all to the various noise conditions and the implications of changes in such conditions. There is nothing in the cited portion of Anandakumar, or in Anandakumar as a whole which would suggest the limitations of claim 1.

For these reasons applicants respectfully submit that claim 1, as amended, is not rendered obvious by Anandakumar. Given that claims 2 – 19 include the limitation as discussed, applicants respectfully submit that claims 2 – 19 are, likewise, not rendered obvious by Anandakumar.

In regard to claim 20, applicants respectfully submit that Anandakumar does not disclose or suggest the limitation of automatically determining the prevalent type of noise on a logical channel and selecting a group of burst profiles suited to the type of prevalent noise on said channel. Again, regardless of whether Anandakumar makes obvious that packet loss may be due to noise. Anandakumar does not disclose determining types of noise nor taking specific action based upon a type of noise determined. Applicants respectfully request the Examiner reconsider whether this limitation is rendered obvious by Anandakumar.

Given that claims 21 – 24 depend from claim 20, applicants respectfully submit that claims 21 – 24 are, likewise, not rendered obvious by Anandakumar.

In view of the foregoing, it is respectfully asserted that the claims are now in condition for allowance.

Conclusion

It is believed that this Amendment places the above-identified patent application into condition for allowance. Early favorable consideration of this Amendment is earnestly solicited.

If, in the opinion of the Examiner, an interview would expedite the prosecution of this application, the Examiner is invited to call the undersigned attorney at the number indicated below.


Applicant respectfully requests that a timely Notice of Allowance be issued in this case.

Please charge any additional required fee or credit any overpayment not otherwise paid or credited to our deposit account No. 50-1698.

Respectfully submitted,

THELEN REID & PRIEST, LLP

Dated: 6/16/06


Thomas Van Zandt
Reg. No. 43,219

Thelen Reid & Priest LLP
P.O. Box 640640
San Jose, CA 95164-0640
Tel. (408) 292-5800
Fax. (408) 287-8040